This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1. (Original) A method for indexing data, comprising:

receiving a token;

determining whether a data field associated with the token is a fixed width;

when the data field is a fixed width, designating the token as one for which fixed width sort is to be performed; and

when the data field is a variable length, designating the token as one for which a variable width sort is to be performed.

2. (Original) The method of claim 1, wherein the token is variable width and further comprising:

transforming the variable width token into a fixed width token.

3. (Original) The method of claim 1, further comprising:

performing a fixed width sort on one of dual code paths and a variable width sort on the other of dual code paths.

4. (Original) The method of claim 1, further comprising:

generating a sort key that includes a token type, a token, a document identifier, a document section, and an offset in a document.

5. (Original) The method of claim 1, further comprising:

receiving different sections of a document at different times.

6. (Original) The method of claim 5, wherein the different sections include a context section and an anchor text section.

7. (Original) The method of claim 1, further comprising:

generating sort keys for each token of multiple tokens; and

using the sort keys to create posting lists that simultaneously are ordered by token and by document identifier for each token.

8. (Original) The method of claim 7, further comprising:

using the sort keys to bring together multiple sections of a document.

9. (Original) The method of claim 1, further comprising:

sorting on certain bits of a sort key containing multiple bits.

10. (Original) The method of claim 9, further comprising:

sorting on uppermost bits of the sort key.

11. (Original) A computer system including logic for indexing data, comprising:

receiving a token;

determining whether a data field associated with the token is a fixed width;

when the data field is a fixed width, designating the token as one for which fixed width sort is to be performed; and

when the data field is a variable length, designating the token as one for which a variable width sort is to be performed.

12. (Original) The computer system of claim 11, wherein the token is variable width and wherein the logic further comprises:

transforming the variable width token into a fixed width token.

13. (Original) The computer system of claim 11, wherein the logic further comprises:

performing a fixed width sort on one of dual code paths and a variable width sort on the other of dual code paths.

14. (Original) The computer system of claim 11, wherein the logic further comprises:

generating a sort key that includes a token type, a token, a document identifier, a document section, and an offset in a document.

15. (Original) The computer system of claim 11, wherein the logic further comprises:

receiving different sections of a document at different times.

16. (Original) The computer system of claim 15, wherein the different sections include a context section and an anchor text section.

17. (Original) The computer system of claim 11, wherein the logic further comprises:

generating sort keys for each token of multiple tokens; and

using the sort keys to create posting lists that simultaneously are ordered by token and by document identifier for each token.

18. (Original) The computer system of claim 17, wherein the logic further comprises:

using the sort keys to bring together multiple sections of a document.

19. (Original) The computer system of claim 11, wherein the logic further comprises:

sorting on certain bits of a sort key containing multiple bits.

20. (Original) The computer system of claim 19, wherein the logic further comprises:

sorting on uppermost bits of the sort key.

21. (Currently Amended) An article of manufacture comprising one of hardware logic and a computer readable medium including a program for indexing data, wherein the hardware logic or program causes operations to be performed, the operations comprising:

receiving a token;

determining whether a data field associated with the token is a fixed width;

when the data field is a fixed width, designating the token as one for which fixed width sort is to be performed; and

when the data field is a variable length, designating the token as one for which a variable width sort is to be performed.

22. (Original) The article of manufacture of claim 21, wherein the token is variable width and wherein the operations further comprise:

transforming the variable width token into a fixed width token.

23. (Original) The article of manufacture of claim 21, wherein the operations further comprise:

performing a fixed width sort on one of dual code paths and a variable width sort on the other of dual code paths.

24. (Original) The article of manufacture of claim 21, wherein the operations further comprise:

generating a sort key that includes a token type, a token, a document identifier, a document section, and an offset in a document.

25. (Original) The article of manufacture of claim 21, wherein the operations further comprise:

receiving different sections of a document at different times.

26. (Original) The article of manufacture of claim 25, wherein the different sections include a context section and an anchor text section.

27. (Original) The article of manufacture of claim 21, wherein the operations further comprise:

generating sort keys for each token of multiple tokens; and

using the sort keys to create posting lists that simultaneously are ordered by token and by document identifier for each token.

28. (Original) The article of manufacture of claim 27, wherein the operations further comprise:

using the sort keys to bring together multiple sections of a document.

29. (Original) The article of manufacture of claim 21, wherein the operations further comprise:

sorting on certain bits of a sort key containing multiple bits.

30. (Original) The article of manufacture of claim 29, wherein the operations further comprise:

sorting on uppermost bits of the sort key.